

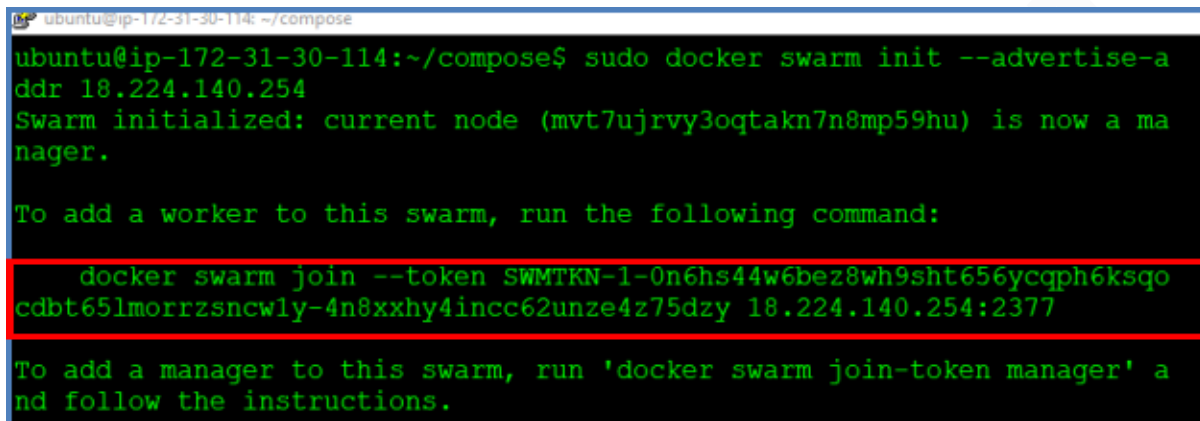


## **Hands-On: Docker Swarm Initialization on Ubuntu**

Since we have already installed Docker in our system, along with that Docker Swarm is already installed. We just need to initialize the Docker Swarm.

**Step 1:** Use the following command to create a new swarm

```
$ sudo docker swarm init --advertise-addr <master IP>
```



```
ubuntu@ip-172-31-30-114: ~/compose
ubuntu@ip-172-31-30-114:~/compose$ sudo docker swarm init --advertise-addr 18.224.140.254
Swarm initialized: current node (mvt7ujrvy3oqtakn7n8mp59hu) is now a manager.

To add a worker to this swarm, run the following command:

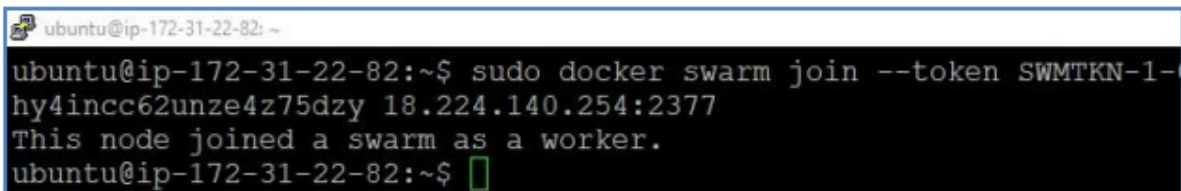
    docker swarm join --token SWMTKN-1-0n6hs44w6bez8wh9sht656ycqph6ksqocdbt65lmorrszsnwly-4n8xxhy4incc62unze4z75dzy 18.224.140.254:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.
```

Copy the token (marked in red) to clipboard

**Step 2:** Now we will start a new session as worker and we will join the swarm that we just created. Paste the copied token shown below.

```
$ sudo <token>
```



```
ubuntu@ip-172-31-22-82: ~
ubuntu@ip-172-31-22-82:~$ sudo docker swarm join --token SWMTKN-1-hy4incc62unze4z75dzy 18.224.140.254:2377
This node joined a swarm as a worker.
ubuntu@ip-172-31-22-82:~$
```

**Step 3:** Now check we will check the node list as the manager

```
$ sudo docker node ls
```

```
ubuntu@ip-172-31-30-114: ~/compose
ubuntu@ip-172-31-30-114:~/compose$ sudo docker node ls
ID                                HOSTNAME                STATUS                AVAILABILITY                MANAGER STATUS
ON
pzb888pme2hlrcccz9ndpe64t      ip-172-31-22-82        Ready                Active
mvt7ujrvy3oqtakn7n8mp59hu *   ip-172-31-30-114      Ready                Active                Leader
ubuntu@ip-172-31-30-114:~/compose$
```

As you can see, worker has joined and statuses of both nodes are ready.

**Step 4:** Follow the commands given below to leave the swarm:

```
$ sudo docker leave --force
```

```
ubuntu@ip-172-31-22-82: ~
ubuntu@ip-172-31-22-82:~$ sudo docker swarm leave --force
Node left the swarm.
ubuntu@ip-172-31-22-82:~$
```

Now that the node left the swarm, let's check the node list as manager and check the status of the nodes.

**Step 5:** To check the node list as manager follow the command given below:

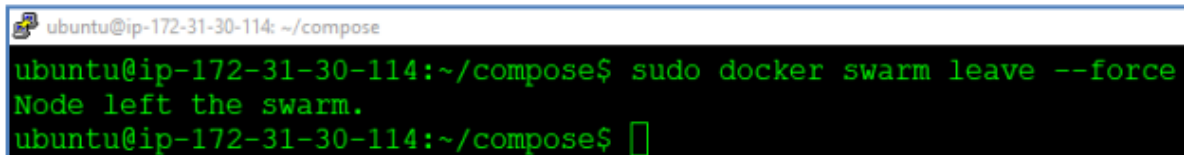
```
$ sudo docker node ls
```

```
ubuntu@ip-172-31-30-114: ~/compose
ubuntu@ip-172-31-30-114:~/compose$ sudo docker node ls
ID                                HOSTNAME                STATUS                AVAILABILITY                MANAGER STATUS
ON
pzb888pme2hlrcccz9ndpe64t      ip-172-31-22-82        Down                Active
mvt7ujrvy3oqtakn7n8mp59hu *   ip-172-31-30-114      Ready                Active                Leader
ubuntu@ip-172-31-30-114:~/compose$
```

As you can, the status of the node that left the swarm is no longer ready.

**Step 6:** To leave the swarm as manager follow the command given below:

```
$ sudo docker swarm leave --force
```

A terminal window screenshot showing the execution of the command to leave the Docker Swarm. The terminal title is "ubuntu@ip-172-31-30-114: ~/compose". The command entered is "sudo docker swarm leave --force", and the output is "Node left the swarm." followed by a new prompt.

```
ubuntu@ip-172-31-30-114: ~/compose
ubuntu@ip-172-31-30-114:~/compose$ sudo docker swarm leave --force
Node left the swarm.
ubuntu@ip-172-31-30-114:~/compose$
```